Fuzzy Weighted Data Mining from Quantitative Transactions with Linguistic Minimum Supports and Confidences

Tzung-Pei Hong, Ming-Jer Chiang, and Shyue-Liang Wang

Abstract

Data mining is the process of extracting desirable knowledge or interesting patterns from existing databases for specific purposes. Most conventional data-mining algorithms identify the relationships among transactions using binary values and set the minimum supports and minimum confidences at numerical values. Linguistic minimum support and minimum confidence values are, however, more natural and understandable for human beings. Transactions with quantitative values are also commonly seen in real-world applications. This paper thus attempts to propose a new mining approach for extracting linguistic weighted association rules from quantitative transactions, when the parameters needed in the mining process are given in linguistic terms. Items are also evaluated by managers as linguistic terms to reflect their importance, which are then transformed as fuzzy sets of weights. Fuzzy operations are then used to find weighted fuzzy large itemsets and fuzzy association rules. An example is given to clearly illustrate the proposed approach.

Keywords: Association rule, data mining, weighted item, fuzzy set, fuzzy ranking, quantitative data.

1. Introduction

Knowledge discovery in databases (KDD) has become a process of considerable interest in recent years as the amounts of data in many databases have grown tremendously large. KDD means the application of non-trivial procedures for identifying effective, coherent, potentially useful, and previously unknown patterns in large databases [15]. The KDD process generally consists of three phases: pre-processing, data mining and post-processing [14, 26]. Among them, data mining plays a critical role to KDD. Depending on the classes of knowledge derived, mining approaches may be classified as finding association rules, classification rules, clustering rules, and sequential patterns [10], among others. It is most commonly seen in applications to induce association rules from transaction data.

Most previous studies have only shown how binary valued transaction data may be handled. Transactions with quantitative values are, however, commonly seen in real-world applications. Srikant and Agrawal proposed a method for mining association rules from transactions with quantitative and categorical attributes [30]. Their proposed method first determined the number of partitions for each quantitative attribute, and then mapped all possible values of each attribute into a set of consecutive integers. It then found large itemsets whose support values were greater than user-specified minimum-support levels. These large itemsets were then processed to generate association rules.

Recently, the fuzzy set theory has been used more and more frequently in intelligent systems because of its simplicity and similarity to human reasoning [36, 37]. The theory has been applied in fields such as manufacturing, engineering, diagnosis, and economics, among others [17, 23, 25, 36]. Several fuzzy learning algorithms for inducing rules from given sets of data have been designed and used to good effect with specific domains [5, 7, 13, 16, 18-21, 29, 31, 32]. Strategies based on decision trees were proposed in [9, 11-12, 27-29, 33-34], and based on version spaces were proposed in [31]. Fuzzy mining approaches were proposed in [8, 22, 24, 35].

Besides, most conventional data-mining algorithms set the minimum supports and minimum confidences at numerical values. Linguistic minimum support and minimum confidence values are, however, more natural and understandable for human beings. In this paper, we thus extend our previous fuzzy mining algorithm [22] for quantitative transactions to the mining problems with linguistic minimum support and minimum confidence values. Also, items may have different importance, which is evaluated by managers or experts as linguistic terms. A novel mining algorithm is then proposed to find weighted linguistic association rules from quantitative transaction data. It first transforms linguistic weighted items, minimum supports, minimum confidences and quantitative transactions into fuzzy sets, and then filters weighted large itemsets out by fuzzy operations. Weighted association rules with linguistic supports and

Corresponding Author: Tzung-Pei Hong is with the Department of Electrical Engineering, National University of Kaohsiung, No. 700, Kaohsiung university Road, Kaohsiung, Taiwan, 811.
E-mail: tphong@nuk.edu.tw

© 2004 TFSA (only used in first page)
The notations are then derived from the weighted large itemsets.

The remaining parts of this paper are organized as follows. Mining association rules is reviewed in Section 2. The notation used in this paper is defined in Section 3. A novel mining algorithm for managing quantitative transactions, linguistic minimum supports and linguistic minimum confidences is proposed in Section 4. An example to illustrate the proposed mining algorithm is given in Section 5. Conclusions and proposal of future work are given in Section 6.

2. Review of Mining Association Rules

The goal of data mining is to discover important associations among items such that the presence of some items in a transaction will imply the presence of some other items. To achieve this purpose, Agrawal and his co-workers proposed several mining algorithms based on the concept of large itemsets to find association rules in transaction data [1-4]. They divided the mining process into two phases. In the first phase, candidate itemsets were generated and counted by scanning the transaction data. If the number of an itemset appearing in the transactions was larger than a pre-defined threshold value (called minimum support), the itemset was considered a large itemset. Itemsets containing only one item were processed first. Large itemsets containing only single items were then combined to form candidate itemsets containing two items. This process was repeated until all large itemsets had been found. In the second phase, association rules were induced from the large itemsets found in the first phase. All possible association combinations for each large itemset were formed, and those with calculated confidence values larger than a predefined threshold (called minimum confidence) were output as association rules.

Srikant and Agrawa then proposed a mining method [30] to handle quantitative transactions by partitioning the possible values of each attribute. Hong et al. proposed a fuzzy mining algorithm to mine fuzzy rules from quantitative data [22]. They transformed each quantitative item into a fuzzy set and used fuzzy operations to find fuzzy rules. Cai et al. proposed weighted mining to reflect different importance to different items [6]. Each item was attached a numerical weight given by users. Weighted supports and weighted confidences were then defined to determine interesting association rules. Yue et al. then extended their concepts to fuzzy item vectors [35]. The minimum supports and minimum confidences set in the above methods were numerical. In this paper, these parameters are expressed in linguistic terms, which are more natural and understandable for human beings.

3. Notation

The notation used in this paper is defined as follows.

- $n$: the total number of transaction data;
- $m$: the total number of items;
- $d$: the total number of managers;
- $u$: the total number of membership functions for item importance;
- $D_i$: the $i$-th transaction datum, $1 \leq i \leq n$;
- $A_j$: the $j$-th item, $1 \leq j \leq m$;
- $h$: the number of fuzzy regions for each item;
- $R_i^j$: the $l$-th fuzzy region of $A_j$, $1 \leq l \leq h$;
- $V_i$: the quantitative value of $A_j$ in $D_i$;
- $f_{ij}$: the fuzzy set converted from $V_i$;
- $f_{ij}^l$: the membership value of $V_i^l$ in Region $R_i^l$;
- $\text{count}_{jl}$: the summation of $f_{ij}^l$, $1 \leq i \leq n$;
- $\text{max-count}_{jl}$: the maximum count value among $\text{count}_{jl}$ values;
- $\text{max}-R_j$: the fuzzy region of $A_j$ with $\text{max-count}_{jl}$;
- $W_i^j$: the transformed fuzzy weight for importance of item $A_j$, evaluated by the $k$-th manager, $1 \leq k \leq d$;
- $W_i^{ave}$: the fuzzy average weight for importance of item $A_j$;
- $\alpha$: the predefined linguistic minimum support value;
- $\beta$: the predefined linguistic minimum confidence value;
- $I_i$: the $i$-th membership function of item importance, $1 \leq i \leq u$;
- $P_i^\alpha$: the fuzzy average weight of all possible linguistic terms of item importance;
- $w^i_{sup}$: the fuzzy weighted support of item $A_j$;
- $w^i_{conf}$: the fuzzy weighted confidence of rule $R$;
- $\text{minsups}$: the transformed fuzzy set from the linguistic minimum support value $\alpha$;
- $\text{wminsups}$: the fuzzy weighted set of minimum supports;
- $\text{minconf}$: the transformed fuzzy set from the linguistic minimum confidence value $\beta$;
- $\text{wminconf}$: the fuzzy weighted set of minimum confidences;
- $C_r$: the set of candidate weighted itemsets with $r$ items;
- $L_r$: the set of large weighted itemsets with $r$ items.

4. The Proposed Algorithm

In this section, we propose a new weighted data-mining algorithm, which can process transaction data with quantitative values and discover fuzzy association rules for linguistic minimum support and confidence values. The fuzzy concepts are used to represent item importance, item quantities, minimum supports and minimum confidences. The proposed mining algorithm
first uses the set of membership functions for importance to transform managers’ linguistic evaluations of item importance into fuzzy weights. The fuzzy weights of an item from different managers are then averaged. Each quantitative value for a transaction item is also transformed into a fuzzy set using the given membership functions. Each attribute used only the linguistic term with the maximum cardinality in the mining process. The number of items was thus the same as that of the original attributes, making the processing time reduced [22]. The algorithm then calculates the weighted fuzzy counts of all items according to the average fuzzy weights of items. The given linguistic minimum support value is then transformed into a fuzzy weighted set. All weighted large 1-itemsets can thus be found by ranking the fuzzy weighted support of each item with the fuzzy weighted minimum support. After that, candidate 2-itemsets are formed from the weighted large 1-itemsets and the same procedure is used to find all weighted large 2-itemsets. This procedure is repeated until all weighted large itemsets have been found. The fuzzy weighted confidences from large itemsets are then calculated to find interesting association rules. Details of the proposed mining algorithm are described below.

The algorithm:

INPUT: A set of \( n \) quantitative transaction data, a set of \( m \) items with their importance evaluated by \( d \) managers, four sets of membership functions respectively for item quantities, item importance, minimum support and minimum confidence, a pre-defined linguistic minimum support value \( \alpha \), and a pre-defined linguistic minimum confidence value \( \beta \).

OUTPUT: A set of weighted fuzzy association rules.

STEP 1: Transform each linguistic term of importance for item \( A_j \), \( 1 \leq j \leq m \), which is evaluated by the \( k \)-th manager into a fuzzy set \( W_{jk} \) of weights, \( 1 \leq k \leq d \), using the given membership functions of item importance.

STEP 2: Calculate the fuzzy average weight \( W_{jave} \) of each item \( A_j \) by fuzzy addition as:

\[
W_{jave} = \frac{1}{d} \sum_{k=1}^{d} W_{jk}
\]

STEP 3: Transform the quantitative value \( V_{ij} \) of each item \( A_j \) in each transaction datum \( D_i \) (\( i = 1 \) to \( n \), \( j = 1 \) to \( m \)), into a fuzzy set \( f_{ij} \) represented as:

\[
\left( \frac{f_{i1}}{R_{j1}}, \frac{f_{i2}}{R_{j2}}, \ldots, \frac{f_{ih}}{R_{jh}} \right)
\]

using the given membership functions for item quantities, where \( h \) is the number of regions for \( A_j \), \( R_{jl} \) is the \( l \)-th fuzzy region of \( A_j \), \( 1 \leq l \leq h \), and \( f_{ij} \) is \( V_{ij} \)'s fuzzy membership value in region \( R_{jl} \).

STEP 4: Calculate the count of each fuzzy region \( R_{jl} \) in the transaction data as:

\[
count_{jl} = \sum_{i=1}^{n} f_{ijl}
\]

STEP 5: Find \( max-count_j = max(\text{count}_{jl}) \), \( j = 1 \) to \( m \), where \( m \) is the number of items. Let \( \text{max}-R_j \) be the region with \( \text{max}-\text{count}_j \) for item \( A_j \). \( \text{max}-R_j \) is then used to represent the fuzzy characteristic of item \( A_j \) in later mining processes for saving computational time.

STEP 6: Calculate the fuzzy weighted support \( \text{wsup}_j \) of each item \( A_j \) as:

\[
\text{wsup}_j = \max - R_j \times W_{jave}
\]

where \( n \) is the number of transactions.

STEP 7: Transform the given linguistic minimum support value \( \alpha \) into a fuzzy set (denoted \( \text{msup} \)) of minimum supports, using the given membership functions for minimum supports.

STEP 8: Calculate the fuzzy weighted support \( \text{wmmsup} \) of the given minimum support value as:

\[
\text{wmmsup} = \text{msup} \times (\text{the gravity of } I_{ave}),
\]

where \( I_{ave} = \frac{1}{u} \sum_{i=1}^{u} I_i \), with \( u \) being the total number of membership functions for item importance and \( I_i \) being the \( t \)-th membership function. \( I_{ave} \) thus represents the fuzzy average weight of all possible linguistic terms of importance.

STEP 9: Check whether the weighted support \( \text{wsup}_j \) of each item \( A_j \) is larger than or equal to the fuzzy weighted minimum support \( \text{wmmsup} \) by fuzzy ranking. Any fuzzy ranking approach can be applied here as long as it can generate a crisp rank. If \( \text{wsup}_j \) is equal to or greater than \( \text{wmmsup} \), put \( A_j \) in the set of large 1-itemsets \( L_1 \).

STEP 10: Set \( r = 1 \), where \( r \) is used to represent the number of items kept in the current large itemsets.

STEP 11: Generate the candidate set \( C_{r+1} \) from \( L_r \) in a way similar to that in the \text{apriori} algorithm [4]. That is, the algorithm first joins \( L_r \) and \( L_r \) assuming that \( r \)-1 items in the two itemsets are the same and the other one is different. It then keeps in \( C_{r+1} \) the itemsets, which have all their sub-itemsets of \( r \) items existing in \( L_r \).
STEP 12: Do the following substeps for each newly formed \((r+1)\)-itemset \(s\) with items \((s_1, s_2, \ldots, s_{r+1})\) in \(C_{r+1}\):
(a) Find the weighted fuzzy set \(W_{fis}\) of \(s\) in each transaction data \(D_i\) as:
\[
W_{fis} = \min_{j=1}^{r+1} (W_{sj} \times f_{isj}),
\]
where \(f_{isj}\) is the membership value of region \(s_j\) in \(D_i\) and \(W_{sj}\) is the average fuzzy weight for \(s_j\).
(b) Calculate the fuzzy weighted support \((wsup_s)\) of itemset \(s\): 
\[
wsup_s = \frac{\sum_{i=1}^{n} W_{fis}}{n},
\]
where \(n\) is the number of transactions.
(c) Check whether the weighted support \((wsup_s)\) of itemset \(s\) is greater than or equal to the fuzzy weighted minimum support \((wminsup)\) by fuzzy ranking. If \(wsup_s\) is greater than or equal to \(wminsup\), put \(s\) in the set of large \((r+1)\)-itemsets \(L_{r+1}\).

STEP 13: If \(L_{r+1}\) is null, then do the next step; otherwise, set \(r = r + 1\) and repeat Steps 11 to 13.

STEP 14: Transform the given linguistic minimum confidence value \(\beta\) into a fuzzy set \((minconf)\) of minimum confidences, using the given membership functions for minimum confidences.

STEP 15: Calculate the fuzzy weighted set \((wminconf)\) of the given minimum confidence value as:
\[
wminconf = minconf \times (the \ gravity \ of \ F^{ave}),
\]
where \(F^{ave}\) is the same as that calculated in Step 9.

STEP 16: Construct the association rules from each large weighted \(q\)-itemset \(s\) with items \((s_1, s_2, \ldots, s_q)\), \(q \geq 2\), using the following substeps:
(a) Form all possible association rules as follows:
\[
s_1 \Lambda s_{j-1} \Lambda s_{j+1} \Lambda \ldots s_q \rightarrow s_j ,
\]
\(j = 1\) to \(q\).
(b) Calculate the fuzzy weighted confidence value \(wconf_R\) of each possible association rule \(R\) as:
\[
wconf_R = \frac{\text{count}_s}{\text{count}_{s-s_j}} \times W_s ,
\]
where 
\[
\text{count}_s = \sum_{i=1}^{n} (\min_{k=1}^{q} f_{is}) \text{ and } W_s = \min_{i=1}^{q} W_{s_i}^{ave}.
\]
(c) Check whether the fuzzy weighted confidence \(wconf_R\) of association rule \(R\) is greater than or equal to the fuzzy weighted minimum confidence \(wminconf\) by fuzzy ranking. If \(wconf_R\) is greater than or equal to \(wminconf\), keep rule \(R\) in the interesting rule set.

STEP 17: For each rule \(R\) with fuzzy weighted support \(wsup_R\) and fuzzy weighted confidence \(wconf_R\) in the interesting rule set, find the linguistic minimum support region \(S_i\) and the linguistic minimum confidence region \(C_j\) with \(wminsup_{i-1} \leq wsup_R < wminsup\) and \(wminconf_{j-1} \leq wconf_R < wminconf_j\) by fuzzy ranking, where:
\[
\text{wminsup}_i = \text{minsup} \times (the \ gravity \ of \ F^{ave}),
\]
\[
wminconf_i = \text{minconf} \times (the \ gravity \ of \ F^{ave}),
\]
\(\text{minsup}_i\) is the given membership function for \(S_i\) and \(wminconf_i\) is the given membership function for \(C_j\). Output rule \(R\) with linguistic support value \(S_i\) and linguistic confidence value \(C_j\).

The rules output after step 17 can then serve as linguistic knowledge concerning the given transactions.

5. An Example

In this section, an example is given to illustrate the proposed data-mining algorithm. This is a simple example to show how the proposed algorithm can be used to generate weighted fuzzy association rules from a set of quantitative transactions. The data set includes six quantitative transactions, as shown in Table 1.

<table>
<thead>
<tr>
<th>TID</th>
<th>ITEMS</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>(A, 4), (B, 4), (E, 9)</td>
</tr>
<tr>
<td>2</td>
<td>(B, 3), (C, 5), (F, 3)</td>
</tr>
<tr>
<td>3</td>
<td>(B, 2), (C, 3), (D, 2), (E, 8)</td>
</tr>
<tr>
<td>4</td>
<td>(A, 7), (C, 7), (E, 9)</td>
</tr>
<tr>
<td>5</td>
<td>(C, 2), (D, 2), (F, 1)</td>
</tr>
<tr>
<td>6</td>
<td>(A, 4), (B, 3), (C, 5), (F, 2)</td>
</tr>
</tbody>
</table>

Each transaction is composed of a transaction identifier and items purchased. There are six items, respectively being \(A, B, C, D, E\) and \(F\), to be purchased. Each item is represented by a tuple (item name, item amount). For example, the first transaction consists of four units of \(A\), four units of \(B\) and nine units of \(E\).

Also assume that the fuzzy membership functions for item quantities are the same for all the items and are shown in Figure 1. In this example, amounts are represented by three fuzzy regions: Low, Middle and High. Thus, three fuzzy membership values are produced for each item amount according to the predefined mem-
membership functions. The importance of the items is evaluated by three managers as shown in Table 2.

Figure 1. The membership functions for item quantities in this example

Table 2. The item importance evaluated by three managers

<table>
<thead>
<tr>
<th>ITEM</th>
<th>MANAGER 1</th>
<th>MANAGER 2</th>
<th>MANAGER 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>Important</td>
<td>Ordinary</td>
<td>Ordinary</td>
</tr>
<tr>
<td>B</td>
<td>Very Important</td>
<td>Important</td>
<td>Important</td>
</tr>
<tr>
<td>C</td>
<td>Ordinary</td>
<td>Important</td>
<td>Important</td>
</tr>
<tr>
<td>D</td>
<td>Unimportant</td>
<td>Unimportant</td>
<td>Very Unimportant</td>
</tr>
<tr>
<td>E</td>
<td>Important</td>
<td>Important</td>
<td>Important</td>
</tr>
<tr>
<td>F</td>
<td>Unimportant</td>
<td>Unimportant</td>
<td>Ordinary</td>
</tr>
</tbody>
</table>

Assume the membership functions for item importance are given in Figure 2.

Figure 2. The membership functions of item importance used in this example

In Figure 2, item importance is divided into five fuzzy regions: Very Unimportant, Unimportant, Ordinary, Important and Very Important. Each fuzzy region is represented by a membership function. The membership functions in Figure 2 can be represented as follows:

Very Unimportant (VU): (0, 0, 0.25),
Unimportant (U): (0, 0.25, 0.5),
Ordinary (O): (0.25, 0.5, 0.75),
Important (I): (0.5, 0.75, 1), and
Very Important (VI): (0.75, 1, 1).

For the transaction data given in Table 1, the proposed mining algorithm proceeds as follows.

Step 1: The linguistic terms for item importance given in Table 2 are transformed into fuzzy sets by the membership functions in Figure 2. For example, item A is evaluated to be important by Manager 1, and can then be transformed as a triangular fuzzy set (0.5, 0.75, 1) of weights. The transformed results for Table 2 are shown in Table 3.

Table 3. The fuzzy weights transformed from the item importance in Table 2

<table>
<thead>
<tr>
<th>ITEM</th>
<th>MANAGER 1</th>
<th>MANAGER 2</th>
<th>MANAGER 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>(0.5, 0.75, 1)</td>
<td>(0.25, 0.5, 0.75)</td>
<td>(0.25, 0.5, 0.75)</td>
</tr>
<tr>
<td>B</td>
<td>(0.75, 1, 1)</td>
<td>(0.5, 0.75, 1)</td>
<td>(0.5, 0.75, 1)</td>
</tr>
<tr>
<td>C</td>
<td>(0.25, 0.5, 0.75)</td>
<td>(0.5, 0.75, 1)</td>
<td>(0.5, 0.75, 1)</td>
</tr>
<tr>
<td>D</td>
<td>(0.25, 0.5)</td>
<td>(0.25, 0.5)</td>
<td>(0, 0.25)</td>
</tr>
<tr>
<td>E</td>
<td>(0.5, 0.75, 1)</td>
<td>(0.5, 0.75, 1)</td>
<td>(0.5, 0.75, 1)</td>
</tr>
<tr>
<td>F</td>
<td>(0.25, 0.5)</td>
<td>(0.25, 0.5)</td>
<td>(0.25, 0.5, 0.75)</td>
</tr>
</tbody>
</table>

Step 2: The average weight of each item is calculated by fuzzy addition. Take Item A as an example. The three fuzzy weights for Item A are respectively (0.5, 0.75, 1), (0.25, 0.5, 0.75) and (0.25, 0.5, 0.75). The average weight is then ((0.5, 0.75, 1) + (0.25, 0.5, 0.75) + (0.25, 0.5, 0.75)) / 3, which is derived as (0.33, 0.58, 0.83). The average fuzzy weights of all the items are calculated, with results shown in Table 4.

Table 4. The average fuzzy weights of all the items

<table>
<thead>
<tr>
<th>ITEM</th>
<th>AVERAGE FUZZY WEIGHT</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>(0.333, 0.583, 0.833)</td>
</tr>
<tr>
<td>B</td>
<td>(0.583, 0.833, 1)</td>
</tr>
<tr>
<td>C</td>
<td>(0.417, 0.667, 0.917)</td>
</tr>
<tr>
<td>D</td>
<td>(0, 0.167, 0.417)</td>
</tr>
<tr>
<td>E</td>
<td>(0.5, 0.75, 1)</td>
</tr>
<tr>
<td>F</td>
<td>(0.083, 0.333, 0.583)</td>
</tr>
</tbody>
</table>

Step 3: The quantitative values of the items in each transaction are represented by fuzzy sets. Take the first item in Transaction 1 as an example. The amount ‘4’ of A is converted into the fuzzy set (0.4/A.Low + 0.6/A.Middle) using the given membership functions (Figure 1). The step is repeated for the other items, and the results are shown in Table 5, where the notation item.term is called a fuzzy region.

Step 4: The scalar cardinality of each fuzzy region in the transactions is calculated as the count value. Take the fuzzy region A.Low as an example. Its scalar cardinality = (0.4 + 0 + 0 + 0 + 0 + 0.4) = 0.8. The step is repeated for the other regions, and the results are shown in Table 6.

Step 5: The fuzzy region with the highest count among the three possible regions for each item is found. Take item A as an example. Its count is 0.8 for Low, 2.0 for Middle, and 0.2 for High. Since the count for Middle
is the highest among the three counts, the region Middle is thus used to represent the item A in later mining processes. The number of item regions is thus the same as that of the original items, making the processing time reduced. This step is repeated for the other items. Thus, “Low” is chosen for B, “Middle” is chosen for C, “Low” is chosen for D, “High” is chosen for E and “Low” is chosen for F.

Table 5. The fuzzy sets transformed from the data in Table 1

<table>
<thead>
<tr>
<th>TID</th>
<th>FUZZY SETS</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>((0.4 + 0.6) \cdot (0.4 + 0.6) \cdot (0.4 + 0.6) \cdot (0.4 + 0.6))</td>
</tr>
<tr>
<td>2</td>
<td>((0.6 + 0.4) \cdot (0.2 + 0.8) \cdot (0.6 + 0.4) \cdot (0.2 + 0.8))</td>
</tr>
<tr>
<td>3</td>
<td>((0.8 + 0.2) \cdot (0.8 + 0.2) \cdot (0.8 + 0.2) \cdot (0.8 + 0.2))</td>
</tr>
<tr>
<td>4</td>
<td>((0.4 + 0.6) \cdot (0.4 + 0.6) \cdot (0.4 + 0.6) \cdot (0.4 + 0.6))</td>
</tr>
<tr>
<td>5</td>
<td>((0.8 + 0.2) \cdot (0.8 + 0.2) \cdot (0.8 + 0.2) \cdot (0.8 + 0.2))</td>
</tr>
<tr>
<td>6</td>
<td>((0.4 + 0.6) \cdot (0.4 + 0.6) \cdot (0.4 + 0.6) \cdot (0.4 + 0.6))</td>
</tr>
</tbody>
</table>

Table 6. The counts of the fuzzy regions

<table>
<thead>
<tr>
<th>ITEM</th>
<th>COUNT</th>
<th>ITEM</th>
<th>COUNT</th>
<th>ITEM</th>
<th>COUNT</th>
<th>ITEM</th>
<th>COUNT</th>
</tr>
</thead>
<tbody>
<tr>
<td>A.Low</td>
<td>0.8</td>
<td>C.Low</td>
<td>1.8</td>
<td>E.Low</td>
<td>0</td>
<td>A.Middle</td>
<td>2.0</td>
</tr>
<tr>
<td>A.Middle</td>
<td>2.0</td>
<td>C.Middle</td>
<td>3.0</td>
<td>E.Middle</td>
<td>1.4</td>
<td></td>
<td></td>
</tr>
<tr>
<td>A.High</td>
<td>0.2</td>
<td>C.High</td>
<td>0.2</td>
<td>E.High</td>
<td>1.6</td>
<td></td>
<td></td>
</tr>
<tr>
<td>B.Low</td>
<td>2.4</td>
<td>D.Low</td>
<td>1.6</td>
<td>F.Low</td>
<td>2.4</td>
<td></td>
<td></td>
</tr>
<tr>
<td>B.Middle</td>
<td>1.6</td>
<td>D.Middle</td>
<td>0.4</td>
<td>F.Middle</td>
<td>0.6</td>
<td></td>
<td></td>
</tr>
<tr>
<td>B.High</td>
<td>0</td>
<td>D.High</td>
<td>0</td>
<td>F.High</td>
<td>0</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Figure 3. The membership functions of minimum supports

Also assume the given linguistic minimum support value is “Low”. It is then transformed into a fuzzy set of minimum supports, \((0, 0.25, 0.5)\), according to the given membership functions in Figure 3.

Step 8: The fuzzy average weight of all possible linguistic terms of importance in Figure 3 is calculated as:

\[ F^{ave} = \left[(0, 0, 0.25) + (0, 0.25, 0.5) + (0.25, 0.5, 0.75) + (0.75, 1, 1)\right] / 5 \]

\[ = (0.3, 0.5, 0.7). \]

The gravity of \(F^{ave}\) is then \((0.3 + 0.5 + 0.7) / 3\), which is 0.5. The fuzzy weighted set of minimum supports for “Low” is then \((0, 0.25, 0.5) \cdot 0.5\), which is \((0, 0.125, 0.25)\).

Table 7. The fuzzy weighted supports of all the items

<table>
<thead>
<tr>
<th>ITEM</th>
<th>FUZZY WEIGHTED SUPPORT</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>(0.111, 0.194, 0.278)</td>
</tr>
<tr>
<td>B</td>
<td>(0.233, 0.333, 0.4)</td>
</tr>
<tr>
<td>C</td>
<td>(0.208, 0.333, 0.458)</td>
</tr>
<tr>
<td>D</td>
<td>(0, 0.044, 0.111)</td>
</tr>
<tr>
<td>E</td>
<td>(0.133, 0.2, 0.267)</td>
</tr>
<tr>
<td>F</td>
<td>(0.033, 0.133, 0.233)</td>
</tr>
</tbody>
</table>

Figure 3. The set of weighted large 1-itemsets for this example

Table 8. The set of weighted large 1-itemsets for this example

<table>
<thead>
<tr>
<th>ITEMSET</th>
<th>COUNT</th>
</tr>
</thead>
<tbody>
<tr>
<td>A.Middle</td>
<td>2.0</td>
</tr>
<tr>
<td>B.Low</td>
<td>2.2</td>
</tr>
<tr>
<td>C.Middle</td>
<td>3.0</td>
</tr>
<tr>
<td>E.High</td>
<td>1.6</td>
</tr>
<tr>
<td>F.Low</td>
<td>2.4</td>
</tr>
</tbody>
</table>
Step 10: $r$ is set at 1, where $r$ is used to store the number of items kept in the current itemsets.

Step 11: The candidate set $C_2$ is first generated from $L_2$ as follows: $(A.Middle, B.Low)$, $(A.Middle, C.Middle)$, $(A.Middle, E.High)$, $(A.Middle, F.Low)$, $(B.Low, C.Middle)$, $(B.Low, E.High)$, $(B.Low, F.Low)$, $(C.Middle, E.High)$, $(C.Middle, F.Low)$, and $(E.High, F.Low)$.

Step 12: The following substeps are done for each newly formed candidate itemset in $C_2$.

(a) The weighted fuzzy set of each candidate 2-itemset in each transaction data is first calculated. Here, the minimum operator is used for intersection. Take $(A.Middle, B.Low)$ as an example. The membership values of $A.Middle$ and $B.Low$ for transaction 1 are 0.6 and 0.4, respectively. The average fuzzy weight of item $A$ is $(0.333, 0.583, 0.833)$ and the average fuzzy weight of item $B$ is $(0.583, 0.833, 1)$ from Step 2. The weighted fuzzy set for $(A.Middle, B.Low)$ in Transaction 1 is then calculated as: \[ \min(0.6 \times (0.333, 0.583, 0.833), 0.4 \times (0.583, 0.833, 1)) = \min((0.2, 0.35, 0.5), (0.233, 0.333, 0.4)) = (0.2, 0.333, 0.4). \] The results for all the transactions are shown in Table 9.

(b) The fuzzy weighted count of each candidate 2-itemset in $C_2$ is calculated. Results for this example are shown in Table 10.

(c) The fuzzy weighted support of each candidate 2-itemset is compared with the fuzzy weighted minimum support by fuzzy ranking. As mentioned above, assume the gravity ranking approach is adopted in this example. $(A.Middle, C.Middle)$, $(A.Middle, E.High)$ and $(B.Low, C.Middle)$ are then found to be large weighted 2-itemsets. They are then put in $L_2$.

Step 13: Since $L_2$ is not null, $r = r + 1 = 2$. Steps 11 to 13 are repeated to find $L_3$. $C_3$ is then generated from $L_2$. In this example, $C_3$ is empty. $L_3$ is thus empty.

Step 14: The given linguistic minimum confidence value is transformed into a fuzzy set of minimum confidences. Assume the membership functions for minimum confidence values are shown in Figure 4, which are similar to those in Figure 3.

The fuzzy weighted support of each candidate 2-itemset is then calculated. Take $(A.Middle, B.Low)$ as an example. The fuzzy weighted count of $(A.Middle, B.Low)$ is $(0.4, 0.683, 0.9)$ and the total number of transaction data is 6. Its fuzzy weighted support is then $(0.4, 0.683, 0.9) / 6$, which is $(0.067, 0.114, 0.15)$. All the fuzzy weighted supports of the candidate 2-itemsets are shown in Table 11.

(b) The fuzzy weighted count of each candidate 2-itemset in $C_2$ is calculated. Results for this example are shown in Table 10.

Table 9. The weighted fuzzy set of $(A.Middle, B.Low)$ in each transaction

<table>
<thead>
<tr>
<th>TID</th>
<th>A.Middle</th>
<th>B.Low</th>
<th>A.Middle ∩ B.Low</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>(0.2, 0.35, 0.5)</td>
<td>(0.233, 0.333, 0.4)</td>
<td>(0.2, 0.333, 0.4)</td>
</tr>
<tr>
<td>2</td>
<td>(0.0, 0.0)</td>
<td>(0.35, 0.5, 0.6)</td>
<td>(0.0, 0.0)</td>
</tr>
<tr>
<td>3</td>
<td>(0.0, 0.0)</td>
<td>(0.466, 0.666, 0.8)</td>
<td>(0.0, 0.0)</td>
</tr>
<tr>
<td>4</td>
<td>(0.266, 0.466, 0.666)</td>
<td>(0.0, 0.0)</td>
<td>(0.0, 0.0)</td>
</tr>
<tr>
<td>5</td>
<td>(0.0, 0.0)</td>
<td>(0.0, 0.0)</td>
<td>(0.0, 0.0)</td>
</tr>
<tr>
<td>6</td>
<td>(0.2, 0.35, 0.5)</td>
<td>(0.35, 0.5, 0.6)</td>
<td>(0.2, 0.35, 0.5)</td>
</tr>
</tbody>
</table>

Table 10. The fuzzy weighted counts of the itemsets in $C_2$

<table>
<thead>
<tr>
<th>ITEMSET</th>
<th>COUNT</th>
<th>ITEMSET</th>
<th>COUNT</th>
</tr>
</thead>
<tbody>
<tr>
<td>$(A.Middle, B.Low)$</td>
<td>$(0.4, 0.683, 0.9)$</td>
<td>$(B.Low, E.High)$</td>
<td>$(0.433, 0.633, 0.8)$</td>
</tr>
<tr>
<td>$(A.Middle, C.Middle)$</td>
<td>$(0.466, 0.823, 1.166)$</td>
<td>$(B.Low, F.Low)$</td>
<td>$(0.116, 0.466, 0.816)$</td>
</tr>
<tr>
<td>$(A.Middle, E.High)$</td>
<td>$(0.466, 0.803, 1.1)$</td>
<td>$(C.Middle, E.High)$</td>
<td>$(0.467, 0.717, 0.967)$</td>
</tr>
<tr>
<td>$(A.Middle, F.Low)$</td>
<td>$(0.066, 0.266, 0.466)$</td>
<td>$(C.Middle, F.Low)$</td>
<td>$(0.199, 0.6, 1)$</td>
</tr>
<tr>
<td>$(B.Low, C.Middle)$</td>
<td>$(0.834, 1.266, 1.567)$</td>
<td>$(E.High, F.Low)$</td>
<td>$(0, 0, 0)$</td>
</tr>
</tbody>
</table>

The fuzzy weighted support of each candidate 2-itemset is then calculated. Take $(A.Middle, B.Low)$ as an example. The fuzzy weighted count of $(A.Middle, B.Low)$ is $(0.4, 0.683, 0.9)$ and the total number of transactions is 6. Its fuzzy weighted support is then $(0.4, 0.683, 0.9) / 6$, which is $(0.067, 0.114, 0.15)$. All the fuzzy weighted supports of the candidate 2-itemsets are shown in Table 11.

Table 11. The fuzzy weighted supports of the itemsets in $C_2$

<table>
<thead>
<tr>
<th>ITEMSET</th>
<th>WEIGHT SUPPORT</th>
<th>ITEMSET</th>
<th>WEIGHT SUPPORT</th>
</tr>
</thead>
<tbody>
<tr>
<td>$(A.Middle, B.Low)$</td>
<td>$(0.067, 0.114, 0.15)$</td>
<td>$(B.Low, E.High)$</td>
<td>$(0.072, 0.106, 0.133)$</td>
</tr>
<tr>
<td>$(A.Middle, C.Middle)$</td>
<td>$(0.078, 0.137, 0.194)$</td>
<td>$(B.Low, F.Low)$</td>
<td>$(0.019, 0.078, 0.136)$</td>
</tr>
<tr>
<td>$(A.Middle, E.High)$</td>
<td>$(0.078, 0.134, 0.183)$</td>
<td>$(C.Middle, E.High)$</td>
<td>$(0.078, 0.119, 0.161)$</td>
</tr>
<tr>
<td>$(A.Middle, F.Low)$</td>
<td>$(0.011, 0.044, 0.078)$</td>
<td>$(C.Middle, F.Low)$</td>
<td>$(0.033, 0.1, 0.167)$</td>
</tr>
<tr>
<td>$(B.Low, C.Middle)$</td>
<td>$(0.139, 0.211, 0.261)$</td>
<td>$(E.High, F.Low)$</td>
<td>$(0, 0, 0)$</td>
</tr>
</tbody>
</table>

Figure 4. The membership functions for minimum confidences

Also assume the given linguistic minimum confidence value is “Middle”. It is then transformed into a fuzzy set of minimum confidences, $(0.25, 0.5, 0.75)$, according to the given membership functions in Figure 4.

Step 15: The fuzzy average weight of all possible linguistic terms of importance is the same as that found in Step 9. Its gravity is thus 0.5. The fuzzy weighted set of minimum confidences for “Middle” is then $(0.25, 0.5, 0.75) \times 0.5$, which is $(0.125, 0.25, 0.375)$. 
Step 16: The association rules from each large itemset are constructed by using the following substeps.

(a) All possible association rules are formed as follows:

- If A.Middle, then C.Middle;
- If C.Middle, then A.Middle;
- If A.Middle, then E.High;
- If E.High, then A.Middle;
- If B.Low, then C.Middle;
- If C.Middle, then B.Low.

(b) The weighted confidence values for the above possible association rules are calculated. Take the first possible association rule as an example. The fuzzy count of A.Middle is 2.0. The fuzzy count of \( A.Middle \cap C.Middle \) is 1.4. The minimum average weight of \( (A.Middle, C.Middle) \) is \((0.333, 0.588, 0.833)\). The weighted confidence value for the association rule “If A.Middle, then C.Middle” is:

\[
\frac{1.4}{2.0} \times (0.333, 0.588, 0.833) = (0.233, 0.412, 0.583).
\]

The weighted confidence values for the other association rules can be similarly calculated.

(c) The weighted confidence of each association rule is compared with the fuzzy weighted minimum confidence by fuzzy ranking. Assume the gravity ranking approach is adopted in this example. Take the association rule "If A.Middle, then C.Middle" as an example. The average height of the weighted confidence for this association rule is \((0.233 + 0.412 + 0.583)/3\), which is 0.409. The average height of the fuzzy weighted minimum confidence for “Middle” is \((0.125 + 0.25 + 0.375)/3\), which is 0.25. Since 0.409 > 0.25, the association rule "If A.Middle, then C.Middle" is thus put in the interesting rule set. In this example, the following six rules are put in the interesting rule set:

1. If a middle number of A is bought then a middle number of C is bought, with a low support and a middle confidence;
2. If a middle number of C is bought then a middle number of A is bought, with a low support and a middle confidence;
3. If a middle number of A is bought then a high number of E is bought, with a low support and a middle confidence;
4. If a high number of E is bought then a middle number of A is bought;
5. If a low number of B is bought then a middle number of C is bought;
6. If a middle number of C is bought then a low number of B is bought.

Step 17: The linguistic support and confidence values are found for each rule R. Take the interesting association rule "If A.Middle, then C.Middle" as an example. Its fuzzy weighted support is \((0.078, 0.137, 0.194)\) and fuzzy weighted confidence is \((0.233, 0.412, 0.583)\). Since the membership function for linguistic minimum support region “Low” is \((0, 0.25, 0.5)\) and for “Middle” is \((0.25, 0.5, 0.75)\), the weighted fuzzy set for these two regions are \((0, 0.125, 0.25)\) and \((0.125, 0.25, 0.375)\). Since \((0, 0.125) < (0.078, 0.137, 0.194) < (0.125, 0.25, 0.375)\) by fuzzy ranking, the linguistic support value for Rule R is then Low. Similarly, the linguistic confidence value for Rule R is High.

The interesting linguistic association rules are then output as:

1. If a middle number of A is bought then a middle number of C is bought, with a low support and a high confidence;
2. If a middle number of C is bought then a middle number of A is bought, with a low support and a middle confidence;
3. If a middle number of A is bought then a high number of E is bought, with a low support and a middle confidence;
4. If a high number of E is bought then a middle number of A is bought, with a low support and a high confidence;
5. If a low number of B is bought then a middle number of C is bought, with a low support and a high confidence;
6. If a middle number of C is bought then a low number of B is bought, with a low support and a middle confidence.

The six rules above are thus output as meta-knowledge concerning the given weighted transactions.

6. Conclusion and Future Work

In this paper, we have proposed a new weighted data-mining algorithm for finding interesting weighted association rules with linguistic supports and confidences from quantitative transactions. Items are evaluated by managers as linguistic terms, which are then transformed and averaged as fuzzy sets of weights. Fuzzy operations including fuzzy ranking are used to find large weighted itemsets and association rules. Compared to previous mining approaches, the proposed one directly manages linguistic parameters, which are more natural and understandable for human beings. In the future, We will attempt to design other fuzzy data-mining models for various problem domains.

7. Acknowledgement

This research was supported by the National Science Council of the Republic of China under contract NSC94-2213-E-390-005.
8. References


